7 Decentralized Navigation

We had seen in the previous chapter that a surprising feature of many real networks is the existence of short paths between most pairs of vertices, and that random graphs share this feature with real networks. Milgram's experiments had discovered the existence of such paths by delivering letters along social links from a sender to a target who did not know each other personally.

While the lengths of the paths in Milgram's experiment is certainly surprising, we have been able to convince ourselves that the randomness in social ties could explain their existence. However, a major puzzle remains, which in fact may be much more challenging to explain: *people were actually able to find these paths.* This fact is remarkable because every person typically has only full knowledge of their own social ties. While we sometimes know a few of our friends' friends, it is unlikely that we have complete knowledge of our two-hop neighborhood - and certainly our knowledge decreases further the more removed our indirect acquaintances.

A *decentralized routing algorithm* is allowed to make local forwarding decisions for a message based only on local information. More specifically, we send one copy of the message from the source (sender) to the destination (target), and the vertex holding the message has to decide which of its neighbors (on the lattice or through a shortcut) to send the message to, based on the following information:

- the positions of the source, destination, and the current vertex in the lattice;
- the history of the message since it left the destination, i.e., all the information used to deliver the message up to the current vertex;
- the position in the lattice of each of the current vertex's neighbors.

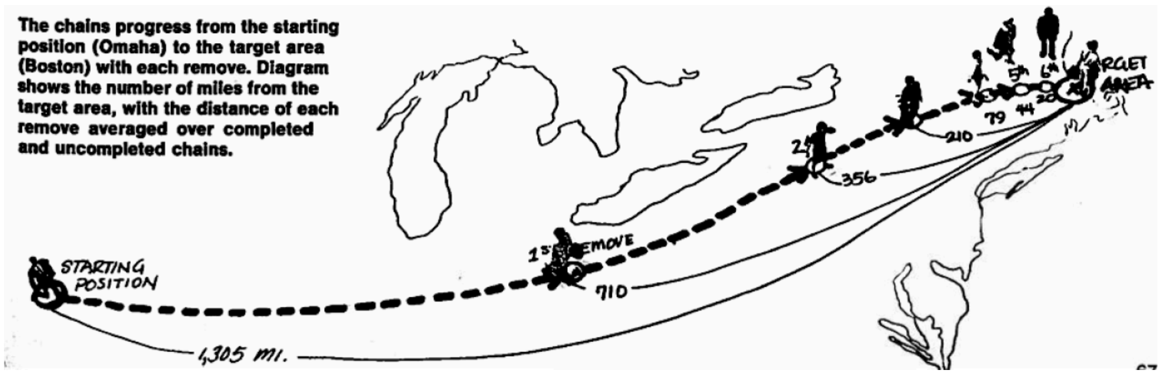

The chains progress from the starting position (Omaha) to the target area (Boston) with each remove. Diagram shows the number of miles from the target area, with the distance of each remove averaged over completed and uncompleted chains.
STARTING POSITION
1ST REMOVE
TARGET AREA
4,305 mi.
710
356
210
79
44
20

Figure 7.1: Image from Milgram's original paper showing the average remaining distance from the current vertex after i steps to the target [?].

The intuition behind such algorithms are clear from Milgram's experiment: individuals clearly used geographic information to zero-in on the target (see Figure 7.1). We are interested in the expected number of iterations that the best possible decentralized routing algorithm can achieve, for a random source s and destination t.

In particular, in this chapter, we discuss a class of models that has been proposed to explain why some networks - and in particular, social networks - appear to be *navigable*, i.e., that it is possible to compute efficient route from sources to destinations where each intermediate vertex only has information about its neighbors. The models we discuss use an underlying geometry to provide a network-wide reference system. This reference system provides a geometric addressing scheme for the destination; also, a vertex knows its own location in the geometry and the locations of its neighbors, and it uses this information to forward a message towards a destination.

These models bring out an interesting puzzle: navigability is not a robust property; rather, it requires a particular way of forming shortcuts between vertices. Why actual social networks exhibit navigability remains an open question, and suggests that we have only started to uncover the principles underlying the structure of such networks, and the mechanisms that give rise to this structure.

7.1 Network Naviation with Watts-Strogatz Networks

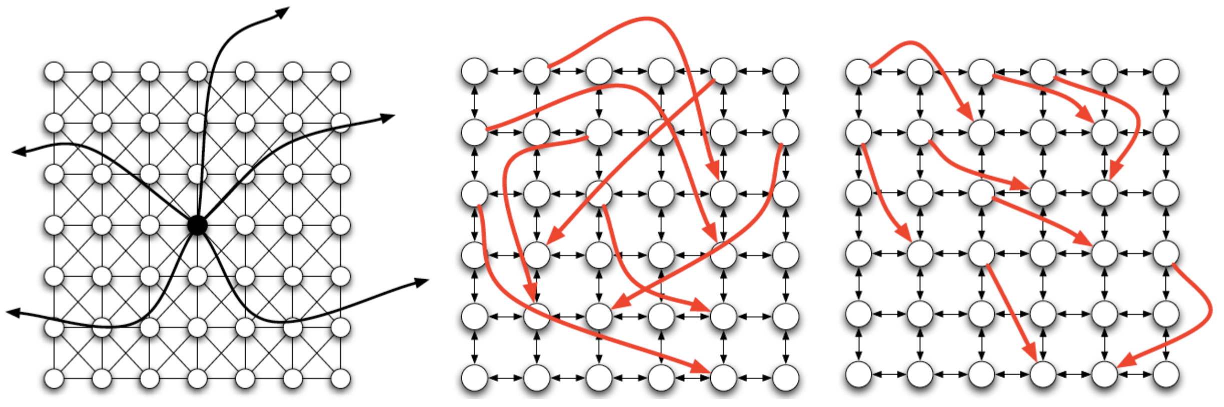
Consider a variant of the Watts-Strogatz model on the square lattice $[n]^2$ with n^2 vertices, and define distance $d(u, v)$ between two vertices u and v as ℓ_1 (i.e., Manhattan) distance. The graph will have *local* edges and remote edges, also known as *shortcuts* as follows: Every vertex v has a local edges to all vertices u with $d(u, v) \leq r$. Every vertex v also has $r \geq 0$ shortcuts, whose endpoints are selected uniformly i.i.d. over the set of other vertices. See Figure 7.2 (a).

Theorem 7.1 (Inefficient Delivery in Watts-Strogatz Grid). *The expected delivery time of any decentralized algorithm is $\Omega(n^{2/3})$.*

Proof:

For the converse, we need to assume that every vertex has r shortcuts. The probability that the current message holder u chooses vertex v as one of its r shortcuts is $1/n-1$ (we do not worry about multiple edges, and simply collapse them into a single edge).

Let $\delta = 2/3$, and let the box B denote the set of vertices within lattice distance n^{δ} of the target t.



(a) The local and remote connections for the highlighted vertex when $r = 2$ and $k = 5$. (b) Small γ correspond to shortcuts with longer reach. (c) Large γ correspond to shortcuts with shorter reach.

Figure 7.2: Images from [?].

There are at most $1 + \sum_{j=1}^{n^\delta} 4j \leq 4n^{2\delta}$ vertices in B.

B is difficult to hit. We first bound the probability that a set of vertices of a given size has at least one shortcut into B. Consider an arbitrary vertex u. Let E_i be the event that at the ith step, the message holder u has a shortcut into B. Let E be the event that any message holder in the first λn^δ steps has a shortcut into B,

$$E = \cup_{i \leq \lambda n^\delta} E_i, \tag{7.1}$$

where $\lambda = {}^{2^{-4}}\!/_{r}$ is a constant.

$$\mathbb{P}\{E_i\} \quad \leq \quad r\frac{|B|}{n^2} \leq \frac{4rn^{2\delta}}{n^2}, \tag{7.2}$$

and by the union bound

$$\mathbb{P}\{E\} \quad \leq \quad \sum_{i \leq \lambda n^\delta} \mathbb{P}\{E_i\} \leq \frac{\lambda n^\delta \cdot 4rn^{2\delta}}{n^2} = 2^2 \lambda r \leq {}^1\!/_4. \tag{7.3}$$

Cost is high if no shortcut into B. If s and t are chosen uniformly at random, then $d(s,t) > \lambda n^\delta$ *a.a.s.* . However, from (7.3), with non-vanishing probability, there is no shortcut into B in the first λn^δ steps. This gives the desired result.

■

7.2 Distance-Dependent Networks

To remedy this, [1] introduced a variant that factors in the distance when considering remote links, as suggested by Milgram's experiment (see Figure 7.1). Again, start with a square lattice with n^2 vertices. The graph will have local edges and remote edges as follows: Every vertex v has a local edges to all vertices u with $d(u, v) \leq r$. Every vertex v also has $r \geq 0$ shortcuts, whose endpoints are selected i.i.d. over the set of other vertices from the distribution

$$\frac{d(u,v)^{-\gamma}}{\sum_{v\neq u} d(u,v)^{-\gamma}}, \tag{7.4}$$

with constant $\gamma \geq 0$. This has a natural interpretation – shortcuts are more likely to occur with "geographically close" vertices. In other words, larger γ concentrate the shortcuts closer to v. Therefore, γ controls how far-reaching the shortcuts are (See Figure 7.2 (b) and (c)).

In the remainder of this chapter, we considered the setting of $r = 1$; larger (constant) r do not affect the results. The model can also be generalized to d-dimensional lattices; in such cases the critical threshold (which we will see below) will occur at $\gamma = d$.

Of course, a lower bound is given by the decentralized algorithm that simply delivers the message through lattice edges on a shortest path, which is guaranteed to succeed in time $O(n)$; ideally we can do better. However, we run up against an immediate negative result – note that when $\gamma = 0$ we recover the Watts-Strogatz model on a square lattice introduced above. In fact, the above theorem and proof can be generalized as follows:

Theorem 7.2 (Inefficient for $0 \leq \gamma < 2$). *The expected delivery time of any decentralized algorithm for $0 \leq \gamma < 2$ is $\Omega(n^\delta)$, with $\delta = (2-\gamma)/3 < 1$.*

Essentially, what the previous theorem says is that for small γ the shortcuts are too random. This makes it difficult to find a shortcut that takes the message close to the target. Although a short path may actually exist, the step-by-step discovery of a decentralized algorithm is unable to find this path, given that it cannot zero in to the target.

In the next theorem, we consider the opposite case, where shortcuts are very local. Unfortunately, this also turns out to be a negative result:

Theorem 7.3 (Inefficient for $\gamma > 2$). *The expected delivery time of any decentralized algorithm is $\Omega(n^\beta)$, where $\beta = (\gamma-2)/(\gamma-1)$.*

Proof:

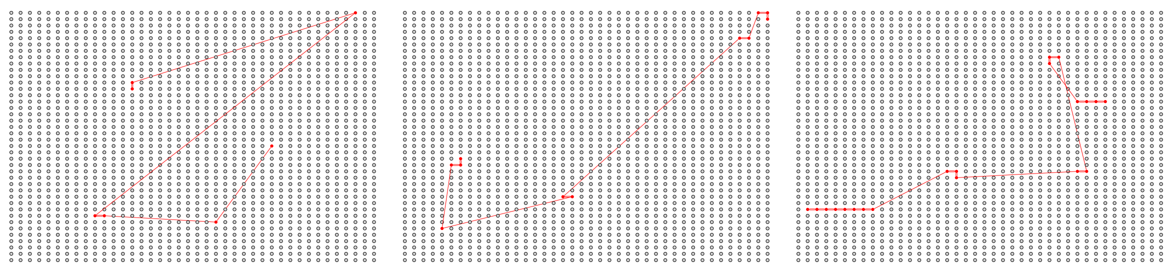
Write $\epsilon = \gamma - 2$, and $\lambda = \min(\epsilon, 1)/8r < 1/2$. As the normalizing constant is at least one,

$$\begin{aligned}
\mathbb{P}\{d(u,v) > d\} &\leq \sum_{j=d+1}^{2n-2} 4j \cdot j^{-\gamma} \\
&= 4 \sum_{j=d+1}^{2n-2} j^{1-\gamma} \\
&\leq \int_d^\infty x^{1-\gamma} dx \\
&\leq \frac{d^{2-\gamma}}{\gamma-2} = d^{-\epsilon}/\epsilon.
\end{aligned} \tag{7.5}$$

Let E_i denote the event that at step i the message reaches a vertex u that has a shortcut of length at least $n^{1-\beta}$. Let $E = \cup_{i \leq \lambda n^\beta} E_i$ be the event that any of the first λn^β vertices has such a shortcut.

By the union bound,

$$\begin{aligned}
\mathbb{P}\{E\} &\leq \sum_{i \leq \lambda n^\beta} \mathbb{P}\{E_i\} \\
&\leq \lambda r n^\beta n^{-\epsilon(1-\beta)}/\epsilon \qquad \text{(combining (7.5) and def. of } E_i) \\
&= \lambda r/\epsilon \leq 1/8. \qquad (\epsilon(1-\beta) = \beta)
\end{aligned} \tag{7.6}$$



(a) A sample shortest path in a lattice-induced topology with $\gamma = 1$. The path is short, but does not follow the geometry of the underlying lattice.

(b) A sample shortest path in a lattice-induced topology with $\gamma = 2$, the critical exponent. The path is short, and most shortcuts make progress towards the destination.

(c) A sample shortest path in a lattice-induced topology with $\gamma = 3$. Note that the shortest path is quite long, even though most hops on the shortest path are in the direction of the destination.

Figure 7.3: Lattice topologies for $r = 1$, $k = 1$ and various γ.

As in the previous proof, we choose s and t uniformly at random, so that they are at typical distance $\Theta(n)$. We want to show that on average, the algorithm requires more than $O(n^\beta)$ steps to complete. It follows from (7.6) that with non-vanishing probability, the message can only find shortcuts of distance less than $n^{1-\beta}$ in the first λn^β steps, for a total progress of order λn. In that case, the message cannot reach the target within the required number of steps.

■

Here, the problem is not that the geometry is not a useful guide towards the target, rather that shortcuts are simply too short to get to the target efficiently.

Is there a sweet-spot in-between? Indeed there is!

Theorem 7.4 (Efficient for $\gamma = 2$)**.** *There exists a decentralized algorithm whose expected delivery time is $O((\log n)^2)$ for n large enough.*

Proof:

We show the result for $r = 1$ and $k = 1$; it is obvious that larger r and k can only help.

The proof is constructive: we define an algorithm to forward a message from the source s to the target t, and show that this algorithm has expected cost of $O((\log n)^2)$.

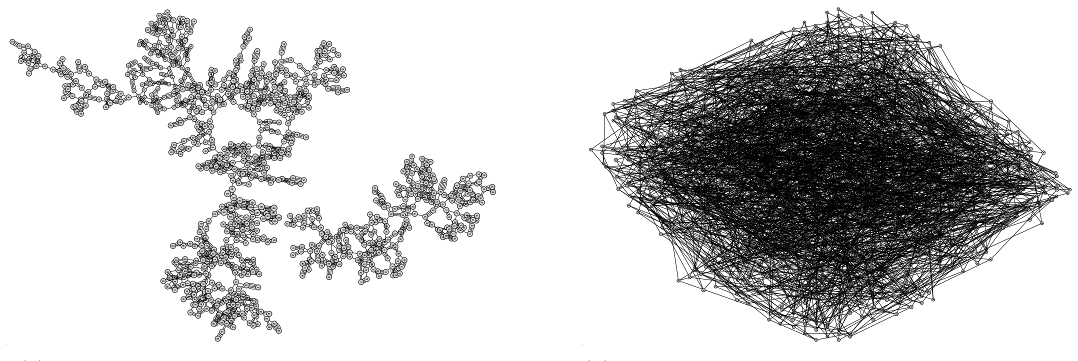
In this algorithm, the current message holder selects the neighbor (on the lattice or through a shortcut) that is closest to the target. Note that it is always possible to make progress towards the target through the lattice; therefore, the algorithm always terminates correctly.

The algorithm is loop-free because, by definition, the message always makes progress. Therefore, by the Principle of Deferred Decisions, we can assume that the shortcuts are generated at a vertex u when that vertex receives the message.

Define the annuli U_j as the set of points at lattice distance in $[2^j + 1, 2^{j+1}]$. Also, the box B_j is the set of points at lattice distance at most 2^j. The algorithm is in phase j while the message is in U_j. Thus, the initial value of j is at most $\log_2 n$.

Consider phase j, $\log\log n \leq j < \log n$. We bound the probability that the algorithm finishes phase j in the next step, and moves into phase $j + 1$. This requires that the message is passed to a vertex in B_j in the next step. The size of the box B_j is at least

$$|B_j| \geq 2^{2j-1}, \tag{7.7}$$



(a) $\alpha = 2$. This network has mostly local links. (b) $\alpha = 0.5$. This network has many long-range links.

Figure 7.4: Tree-induced networks with $n = 1024$, $b = 2$, and different αs.

and the maximum lattice distance between $u \in U_j$ and a vertex in B_j is $2^{j+1} + 2^j < 2^{j+2}$. The probability of hitting a vertex in B_j is at least

$$q_j \geq \frac{|B_j|}{(2^{j+2})^{\gamma} 4 \ln(6n)}, \tag{7.8}$$

where $4\ln(6n)$ is an upper bound of the normalizing constant $\sum_{v \neq u} d(u,v)^{-\gamma}$. Therefore, the message enters B_j with probability at least

$$q_j \geq \frac{2^{2j-1}}{4\ln(6n)2^{2j+4}} = \frac{1}{128\ln(6n)} \tag{7.9}$$

We next bound the expected number of steps in phase j. Let X_j denote this number. Note that X_j is stochastically upper-bounded by a geometric random variable with mean $1/q_j$, and therefore

$$\mathbb{E}[X_j] \leq 128\ln(6n) \tag{7.10}$$

Now the total cost of the algorithm is at most $\sum_{j=0}^{\log n} X_j = O(\log^2 n)$.

■

7.3 Tree-induced Networks

The network model underlying the results on decentralized routing in the previous section is arguably quite peculiar. In particular, there is a priori no reason why a network should "live" in a Euclidean space, which is crucial to allow decentralized routing at $\gamma = 2$. It would therefore be desirable to study a more general class of networks to shed insight.

In a subsequent paper [2], Kleinberg introduces two additional models. The first model uses a balanced tree instead of a lattice as the underlying generator. The probability model to generate shortcuts is the same as in the lattice model, with $d(u,v)$ the tree distance rather than the lattice distance. In contrast to the lattice model, the network on which a message is forwarded only consists of the generated edges, not the underlying tree (as otherwise a path of length $O(\log n)$ always exists on the tree).

The results are similar to those for the lattice case:

Theorem 7.5 (Efficient for $\gamma = 1$)**.** *There exists a tree-induced model with exponent* $\gamma = 1$ *and outdegree of* $\Theta(\log n)$ *such that a decentralized routing algorithm can achieve search time* $O(\log n)$*.*

Theorem 7.6 (Inefficient for $0 \leq \gamma < 1$: no hierarchical clues)**.** *There does not exist a tree-induced model with exponent* $\gamma < 1$ *and outdegree* $O(\log n)$ *such that a decentralized routing algorithm can achieve polylogarithmic search time.*

Theorem 7.7 (Inefficient for $\gamma > 1$: shortcuts too local)**.** *There does not exist a tree-induced model with exponent* $\gamma > 1$ *and outdegree* $O(\log n)$ *such that a decentralized routing algorithm can achieve polylogarithmic search time.*

Bibliography

[1] David Easley and Jon Kleinberg. *Networks, crowds, and markets: Reasoning about a highly connected world.* Cambridge University Press, 2010.

[2] J. Kleinberg. The small-world phenomenon: An algorithmic perspective. In *Proc. 32nd ACM Symposium on Theory of Computing*, 2000.

[3] J. Kleinberg. Small-World Phenomena and the Dynamics of Information. *Advances in Neural Information Processing Systems (NIPS)*, 14, 2001.

[4] Stanley Milgram. The small world problem. *Psychology today*, 2(1):60–67, 1967.